Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks *

Zhenyu Tang and J. J. Garcia-Luna-Aceves
Computer Engineering Department
School of Engineering
University of California
Santa Cruz, CA 95064, USA
ktang,jj@cse.ucsc.edu

Abstract— **A new multichannel MAC protocol called Hop-Reservation Multiple Access (HRMA) for wireless ad-hoc networks (multi-hop packet radio networks) is introduced, specified and analyzed. HRMA is based on simple half-duplex, very-slow frequency-hopping spread spectrum (FHSS) radios and takes advantage of the time synchronization necessary for frequency hopping. HRMA allows a pair of communicating nodes to reserve a frequency hop using a reservation and handshake mechanism that guarantee collision-free data transmission in the presence of hidden terminals. We analyze the throughput achieved in HRMA for the case of a hypercube network topology assuming variable-length packets, and compare it against the multichannel slotted ALOHA protocol, which represents the current practice of MAC protocols in commercial ad-hoc networks based on spread spectrum radios, such as Metricom's Ricochet system. The numerical results show that HRMA can achieve much higher throughput than multichannel slotted ALOHA within the traffic-load ranges of interest, especially when the average packet length is large compared to the duration of a dwell time in the frequency hopping sequence, in which case the maximum throughput of HRMA is close to the maximum possible value.**

I. Introduction

Because of the recent affordability of commercial radios and microprocessor-based controllers, multi-hop packet radio networks (i.e., ad-hoc networks) are likely to play an important role in computer communications. Ad-hoc networks extend packet switching technology into environments with mobile users, can be installed quickly in emergency situations, and are self configurable, which makes them very attractive in many applications, including the seamless extension of the Internet to the wireless, mobile environment.

The unlicensed nature of ISM bands makes them extremely attractive for ad-hoc networks; furthermore, there is widespread availability of commercial, affordable radios for the 915MHz, 2.4GHz and 5.8GHz bands. Accordingly, developing medium access control (MAC) protocols with which the nodes (packet-radios) of ad-hoc networks can share the ISM bands efficiently is critical for the future success of such networks.

In ISM bands, radios must operate using direct sequence spread spectrum (DSSS) or frequency hopping spread spectrum (FHSS)[3]. This paper focuses on the design of an efficient MAC protocol for ad-hoc networks based on FHSS radios operating in ISM bands.

The maximum dwell time on a frequency hop allowed in ISM bands is 400 msec[3], which at 1Mbps allows entire packets to be transmitted within the same frequency hop. On the other hand, keeping the sender and receiver synchronized on the same frequency hops while a packet is being transmitted is not simple when nodes move and data rates are high (1Mbps). Given the FCC regulations for ISM bands and the characteristics of today's COTS radios, the problem of designing MAC protocols that use very-slow frequency hopping (i.e., an entire packet is sent in the same hop) as a combination of time and frequency division multiplexing of the radio channel is very timely. Curiously, there is little work reported on this subject. There are many prior examples of MAC protocols for frequency-hopping radios, which are typically based on applying ALOHA or slotted ALOHA using the same hopping sequence for all nodes or sender- or receiver-oriented code assignments [7][8]. However, these approaches assume that radios hop frequencies within the same packet frequently to achieve code division multiple access (CDMA). IEEE 802.11[1] incorporates a convergent layer that makes the characteristics of the physical layer transparent to the MAC protocol. A concrete example of using very-slow frequency-hopping radios is the MAC protocol used in Metricom's Ricochet wireless network[2], which assumes that each receiver has its own channel (hopping sequence) and makes the sender learn the hopping sequence of the receiver. The sender synchronizes with the receiver's hopping sequence and transmits all its data packet over the same frequency hop at which synchronization occurred. The data packet can last longer than a normal frequency-hop dwell time, which is the hop duration time when there is no data. However, neither [1] nor [2] provides collision-free data transmission in the presence of hidden terminals.

We introduce the Hop-Reservation Multiple Access (HRMA) protocol, which takes advantage of the time-slotting properties of very-slow FHSS. Section II specifies HRMA in detail. HRMA uses a common hopping sequence and permits a pair of nodes to reserve a frequency hop over which they can communicate without interference. A frequency hop is reserved by contention through a request-to-send/clear-to-send exchange between a sender and a receiver. A successful exchange leads to a reservation of a frequency hop, and each reserved hop can remain reserved with a reservation packet from the receiver to the sender, which prevents those nodes that can cause interference from attempting to use the reserved frequency hop. After a hop is reserved, a sender is able to transmit data beyond the normal frequency-hop dwell time on the reserved frequency hop. A common frequency hop is used to permit nodes to synchronize with one another. Section III demonstrates that HRMA guarantees that no data or acknowledgment packets from a source

*This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Grant F30602-97-2-0338

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **1999**	2. REPORT TYPE	3. DATES COVERED **00-00-1999 to 00-00-1999**

4. TITLE AND SUBTITLE **Hp-Reservation Multiple Access (HRMA) for Ad-Hoc Networks**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**8**	

or a receiver collide with any other packets in the presence of hidden terminals. Section IV provides an approximate throughput analysis of HRMA for the case of a hypercube topology, which constitutes the worst-case scenario for the hidden terminal interference, and variable-length packets. The same analysis is presented for the multichannel slotted ALOHA protocol with receiver-oriented channel assignment (ROCA). Section V presents the numerical results of our analysis comparing the two protocols; the results show that HRMA achieves very high throughput for the range of traffic load within which the network is stable, which can be enforced in practice with simple backoff strategies. Section VI presents our conclusions.

II. HRMA Protocol

HRMA is based on a common hopping sequence for the entire network and requires half-duplex slow frequency-hopping radios with no carrier sensing to operate. HRMA can be viewed as a time-slot reservation protocol in which a time slot is also assigned a separate frequency channel.

A. Organizing Time and Frequencies

HRMA uses one of the L available frequencies, which we denote by f_0, as a dedicated synchronizing channel on which its nodes exchange synchronization information. The rest of the frequencies are further divided into $M = \lfloor (L-1)/2 \rfloor$ frequency pairs $(f_i, f_i^*), i = 1, 2, ..., M$. Hence, the length of the hopping sequence is M. For any i, f_i is used for sending or receiving hop-reservation (HR) packets, request-to-send (RTS) packets, clear-to-send (CTS) packets, and data packets while f_i^* is used for sending or receiving acknowledgments to data packets sent on f_i.

As in any MAC protocol operating with FHSS radios, time in HRMA is slotted. Each HRMA slot consists of one synchronizing period, one HR period, one RTS period and one CTS period, each of which is used to exclusively send or receive the synchronizing packet, the HR packet, the RTS packet, and the CTS packet, respectively. Each slot is assigned a frequency hop, which is one of the M frequency hops in the common hopping sequence. All the nodes that are not transmitting or receiving data packets, which we call idle nodes, hop together. All idle nodes must hop to the synchronizing frequency f_0 and exchange synchronizing messages during the synchronizing period of each slot. During the HR, RTS and CTS periods of each slot, however, all idle nodes must dwell on the common frequency hop assigned to each slot. We call the frequency hop assigned to the current slot the current frequency hop.

For synchronization purposes, a special slot called synchronizing slot is defined that is of the same size as a normal slot. All idle nodes must dwell on the synchronizing frequency f_0 during the synchronizing slot to exchange synchronization messages. The exchange of synchronization messages on f_0 in synchronizing slot or synchronizing period allows nodes to synchronize with one another, i.e., to agree on the beginning of a frequency hop in the common hopping sequence and the current frequency hop.

The synchronizing slot followed by the M consecutive normal slots, which pass through all the M frequencies in the common hopping sequence makes up an HRMA frame. Fig. 1 shows

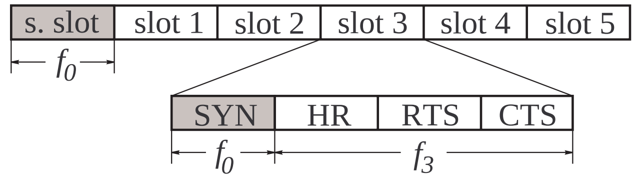

Fig. 1. Structure of HRMA slot and frame

an example of the HRMA frame, where there are five frequencies in the hopping sequence, and thus, six slots in a frame.

B. Synchronizing Nodes to a Common Hopping Sequence

When a new node becomes operational, it must listen to the synchronizing channel for a time period long enough to gather the synchronization information about hopping pattern and timing of the system so that it can get synchronized with the system. If the new node does not detect any synchronization information during that time, it finds an empty system. The new node can broadcast its own synchronization information and create a new one-node system. A new node can easily join or create a system with HRMA because the synchronization information is repeated in every HRMA slot. Hence, nodes in the same connected component of a network, which we call group, are synchronous with each other. In contrast, nodes from different groups are disconnected and asynchronous.

Let the length of a HRMA slot and the synchronizing period of a normal HRMA slot be η and η_s, respectively. It can be seen from Fig. 2 that the dwell time of f_0 at the beginning of each frame is $\eta + \eta_s$. Because the synchronizing period is repeated at the beginning of each HRMA slot, there must be at least one f_0 synchronizing period of length η_s within any interval of length $\eta+\eta_s$. Therefore, any two nodes from disconnected groups must always have at least two overlapping time periods of length η_s on f_0 within any time period equal to the duration of an HRMA frame no matter how large the timing offset between the different groups is. Fig. 2 shows the worst case overlapping time between asynchronous groups. Therefore, HRMA allows different groups to merge. A synchronization protocol based on a listen before transmit policy for beacon packets similar to that advocated in IEEE 802.11[1] can be used in the synchronizing periods and synchronizing slots. However, it would be difficult for asynchronous groups to merge in 802.11 networks. In the rest of this paper, we assume that all nodes are synchronized.

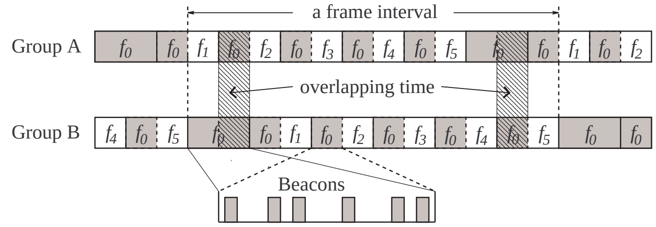

Fig. 2. Worst case overlapping time on synchronizing frequency

C. Accessing and Reserving Hops

Assuming that nodes are able to synchronize according to a common hopping sequence, the rest of HRMA's operation pertains to the way in which nodes access and reserve specific frequency hops. To simplify our analysis, we assume a non-persistent policy for hop reservations; persistent versions of HRMA are also possible.

When an idle node receives a data packet to transmit before the RTS period of a given slot has started, the node backs off if the HR period contains an HR packet. The back-off time is random and is a multiple of the HRMA slot time, so that the node is ready to attempt transmission at the beginning of a slot after the back-off time elapses. Otherwise, if there is no HR packet claiming the slot, the node sends an RTS to the intended receiver and waits for the CTS. Whenever a node receives an RTS intended for it, it sends a CTS back to the source in the CTS period of the same slot and stays in the same frequency hop (instead of hopping to the next frequency) waiting for the data packet. If the node receives no CTS from the receiver in the CTS period, it backs off a random number of slots and tries to send its RTS again in another slot. If the source node receives a CTS for him from the receiver, the source and the receiver have reserved the current frequency hop and the source is able to transmit its data packet after the CTS period. The source and receiver dwell on the same reserved frequency hop during the entire data transmission.

When an idle node receives a data packet to transmit after the RTS period of a given slot has started, the node simply backs off. This is done because such a node is unable to request a hop in the current slot anymore.

After the CTS period of a slot, all nodes that are not transmitting or receiving data packets hop to f_0 and dwell on f_0 for a time period of length η_s to exchange synchronization information, and then hop to the next frequency hop of the M frequencies in the common hopping pattern.

A data packet transmitted in HRMA can be of any length and a node can send multiple data packets as well. However, since HRMA operates in the ISM bands, a data packet or packet train cannot exceed the maximum hop dwell time allowed by the FCC. When the data that need to be exchanged between sender and receiver require multiple HRMA frames for their transmission, the sender notifies the receiver in the header of the data packet and the receiver sends an HR packet during the HR period of the same slot of the next frame. This informs the neighbors of the receiver that they cannot attempt to use the frequency hop occupied by sender and receiver. When the sender receives the HR from the receiver, it sends an RTS to jam any possible RTSs addressed to its own neighbors, which may not hear the receiver. Thus, without further contention, the frequency hop remains reserved by the sender and receiver for the following HRMA frame. Both sender and receiver keep silent in the CTS period of the slot, and more data are transmitted after that over the same reserved frequency hop. The hop remains reserved in a similar fashion, until the sender relinquishes it.

After the source sends a data packet, it hops to the corresponding acknowledgment frequency, and the receiver sends an acknowledgment packet back to the source on the acknowledgment frequency.

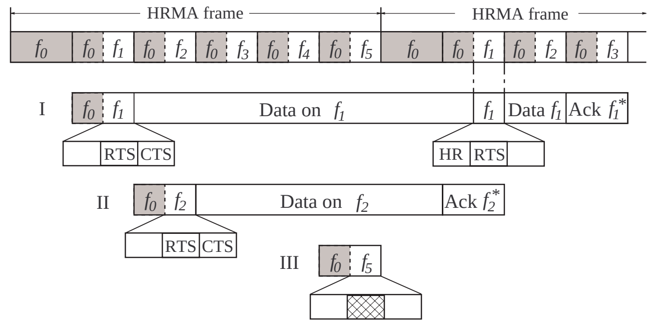

Fig. 3. HRMA basic operations

Fig. 3 shows the different cases for access and reservation of hops in HRMA, namely, Case I: a successful reservation for data longer than a frame; Case II: a successful reservation for data shorter than a frame; and Case III: an unsuccessful reservation.

A more efficient variant of HRMA allows the data including piggybacked acknowledgment to flow in both directions and establishs a duplex data pipe between a pair of nodes, with one node transmitting on f_i and the other on f_i^*. With this approach, the same hop reservation procedure is needed whenever the data in either direction last longer than an HRMA frame.

The pseudo code in Fig. 4 presents the specification of HRMA. We note that the mechanism used to contend for and reserve frequency hops in HRMA is similar in complexity to such simple MAC protocols as FAMA[4][5] and MACAW[6].

III. Correctness of HRMA

The following theorem proves that HRMA eliminates hidden-terminal interference problems. To prove this theorem, we assume that all nodes are synchronized, that there is no capture effect on any channel, and that any overlap of transmissions at any receiver on any channel causes all packets to be lost. Links are bidirectional, which is a requirement that stems from RTS/CTS exchange.

A neighbor of a node A is a node that has a link to A. All the neighbors of node A are denoted by the set $N(A)$. We call a time period equal to the duration of an HRMA frame a frame interval.

Theorem: HRMA guarantees that no data or acknowledgement packet collides with any other packet in the presence of hidden terminals.

Proof: If no RTS is successful, then no data packet or acknowledgment packet is sent and thus no data or acknowledgment packet is involved in any collision.

If a destination node D successfully receives an RTS from a source node S on frequency hop f_k in slot m, it must be true that no node other than S in $N(D)$ is transmitting on f_k in the RTS period of slot m; otherwise, there will be a collision of RTSs at the destination D. Therefore, no other node in $N(D)$ can be a source node on f_k during the following frame interval. However, note that any other node in $N(D)$ can be or become a successful destination on hop f_k if it is not in $N(S)$. It must also be true that no node other than D in $N(S)$ can receive a correct RTS for it in slot m; for otherwise the RTS from S would interfere with it. Accordingly, no nodes other than D in $N(S)$ can be

```
Variable Definitions
    $SL$ = Slot Length
    $FL$ = Frame Length
    $PD$ = Packet Detected
    $TE$ = Timer Expired
    $LD$ = Local Data
    $Tprop$ = Maximum Propagation Delay
    $Tproc$ = Processing Delay

Procedure START()
Begin
    Timer $\leftarrow 3 \times FL$;
    While ($\overline{PD} \wedge \overline{TE}$) listen;
    If($PD$)
    Then Begin
        Receive packet;
        DO CACE of (received packet type)
        Begin
            Synchronizing packet:
                Accept the synchronizing parameters;
                Call PASSIVE();
            Default:
                Call START();
        End
    End
    Else Begin
        Set the synchronizing parameters and send synchronizing packet;
        Call PASSIVE();
    End
End

Procedure ACCESS ()
Begin
    Send RTS to destination in RTS period;
    Listen during CTS period;
    If (CTS received for the RTS sent) Then call XMIT ();
    Else call BACKOFF ();
End

Procedure PASSIVE()
Begin
    DO CASE of (event type)
    Begin
        Beginning of Synch period:
            Call SYNCH ();
        During HR period:
            Listen;
        End of HR period:
            If ($LD$ in Synch or HR period $\wedge \overline{PD}$ in HR period)
            Then call ACCESS ();
        During RTS period:
            Listen;
        Beginning of CTS period:
            If (RTS received $\wedge$ Destination ID = Local ID)
            Then Begin
                send CTS to source;
                Call RECEIVE ();
            End
        End of CTS period:
            If (($LD$ in RTS or CTS period) $\vee$ ($LD \wedge PD$ in HR period))
            Then call BACKOFF ();
            Else call PASSIVE ();
    End
End

Procedure RECEIVE ()
Begin
    $More\_Packet \leftarrow 1$;
    HR_Timer $\leftarrow FL$;
    While (HR_Timer not expired $\wedge More\_Packet$)
    Begin
        Timer $\leftarrow 2 \times Tprop + Tproc$;
        While ($\overline{TE} \wedge \overline{PD}$) wait;
        If ($TE$)
        Then Begin
            If ($LD$) Then call BACKOFF ();
            Else call PASSIVE ();
        End
        Else Begin
            Receive Packet;
            DO CASE of (received packet type)
            Begin
                DATA:
                    If (Destination ID = Local ID)
                    Then Begin
                        Pass packet to upper layer;
                        If (last packet)
                        Then Begin
                            $More\_Packet \leftarrow 0$;
                            Hop to the corresponding ack frequency;
                            Send Ack to source;
                        End
                    End
                    Else Begin
                        If ($LD$) Then call BACKOFF ();
                        Else call PASSIVE ();
                    End
                Default:
                    If ($LD$) Then call BACKOFF ();
                    Else call PASSIVE ();
            End
        End
    End
    If ($\overline{More\_Packet}$)
    Then Begin
        If ($LD$) Then call BACKOFF ();
        Else call PASSIVE ();
    End
    Else Begin
        Send HR packet in the next HR period;
        Listen during RTS and CTS period;
        Call RECEIVE () at the end of CTS period;
    End
End

Procedure SYNCH()
Begin
    Hop to $f_0$;
    While (in Synch period) do synch information exchange;
    Hop to the next frequency according the hopping pattern;
    Return to calling procedure;
End

Procedure XMIT ()
Begin
    If (data length $> FL$)
    Then Begin
        Construct a data packet of length $FL$ by taking part of the data;
        $More\_Packet$ field in packet header $\leftarrow 1$;
    End
    Else Begin
        Put all the data in a packet;
        $More\_Packet$ field in packet header $\leftarrow 0$;
    End
    Transmit data packet;
    Timer $\leftarrow$ delay till the end of the next HR period;
    If (No more data) Then hop to the corresponding ack frequency;
    While ($\overline{TE} \wedge \overline{PD}$) Listen;
    If ($PD$)
    Then Begin
        Receive packet;
        DO CASE of (event type)
        Begin
            More data $\wedge$ HR received for this link:
                Remove the data sent from data buffer;
                Send RTS;
                Call XMIT () at beginning of next slot;
            No more data $\wedge$ Ack received for this packet:
                Remove the data sent from data buffer;
                If ($LD$) call BACKOFF();
                Else call PASSIVE();
            Default:
                Call BACKOFF();
        End
    End
    Else call BACKOFF ();
End

Procedure BACKOFF()
Begin
    Calculate maximum number of backoff slots, m,
    according to some backoff algorithm;
    Timer $\leftarrow RANDOM(0, m \times SL)$;
    While (TRUE) Do
    Begin
        DO CASE of (event type)
        Begin
            Beginning of Synch period:
                Call SYNCH();
            During HR period:
                Listen;
            End of HR period:
                If ($TE \wedge \overline{PD}$ in HR period) Then call ACCESS ();
            During RTS period:
                Listen;
            Beginning of CTS period:
                If (RTS received $\wedge$ Destination ID = Local ID)
                Then Begin
                    Send CTS to source;
                    Call RECEIVE ();
                End
                Else listen during CTS period;
            End of CTS period:
                If ($TE \wedge PD$ in HR period) Then call BACKOFF ();
        End
    End
End
```

Fig. 4. HRMA Specification

or become successful destinations on hop f_k during the following frame interval, however, they can be or become successful sources on f_k if they are not in $N(D)$. As a result, during the following frame interval, S is the only source on f_k in $N(D)$ and D is the only successful destination on f_k in $N(S)$. Therefore, the CTS from D and data packet from S are collision free.

If the data packet lasts longer than a frame, the destination sends an HR in the same slot (slot m), and thus, on the same frequency hop (f_k) of the next frame, which prevents any other node in $N(D)$ from sending an RTS on f_k in slot m and becoming a source node. HR is collision free at S, because R is the only destination on f_k in $N(S)$. After S receives an HR, it sends an RTS on f_k in slot m, and this prevents any other node in $N(S)$ from correctly receiving any possible RTS on f_k directed for that node and becoming a destination. Therefore it is still true that S is the only source on f_k in $N(D)$ and D is the only successful destination on f_k in $N(S)$ during another frame interval. Also note that nodes in $N(S)$ but not in $N(D)$ can become successful sources and nodes in $N(D)$ but not in $N(S)$ can become successful destinations on f_k during the following frame interval. Therefore, again a data packet from S will be collision free in any subsequent frame interval, until the end of the data.

The acknowledgment packet for a data packet is sent on a different frequency of the corresponding frequency pair, f_k^*; therefore, an acknowledgment packet could only collide with other acknowledgment packets. However, as stated above, no two successful destinations exist in the neighborhood of any successful source on the same frequency hop, which implies no acknowledgment packet can collide with any other acknowledgment packet.

Therefore, it follows that HRMA guarantees that data and acknowledgment packets are free of collision in the presence of hidden terminals. Q.E.D.

IV. Comparative Throughput Analysis

A. System Model and Assumptions

For simplicity, we assume a symmetric hypercube network topology, in which each node has N neighbors and the neighbors of the same node are hidden from each other. All links are bidirectional or symmetrical. This type of topology constitutes the worst-case scenario for the hidden terminal interference, and assuming the same number of neighbors per node permits us to focus on any one node to analyze the throughput of the system. Radios are half-duplex and each radio can only tune on to one frequency at a time. Throughput is defined as the average uti-

lization of the receiver (or transmitter) per node, i.e., the probability that each node is receiving (or transmitting) data packets successfully. Because we assume half-duplex radios, the maximum throughput of any MAC protocol is 0.5.

We assume new or retransmitted data packets arrive at each node according to Poison process with average arrival rate λ packets per second. Each node has exactly one buffer for the data packets. The destination of any data packet from each node is uniformly distributed among all its neighbors. All the nodes are synchronized with slot size equal to η. The traffic load normalized to slot size is denoted by $G = \lambda\eta$. To simplify our analysis and to focus on the MAC protocol, we ignore any propagation delay, guard time or any processing time. They can be easily taken into account if necessary, and in ad-hoc networks operating in ISM bands, propagation and processing delays are far smaller than packet lengths. Since IP packets have variable sizes, we are only interested in variable-length data packets. We assume that any data packet is transmitted at the beginning of a slot and ends at the end of a slot; therefore the size of the data packet δ is a multiple of the slot size. We further assume that δ follows a geometric distribution with an average size of d slots, which implies that the probability that a data packet ends at the end of a slot is $q = 1/d$. We also denote by $p = 1 - q$ the probability that a data packet does not end during a slot.

The channels are assumed to be error free and have no capture effect, so that collision of packets is the only source of errors, and more than one packet overlapped on the same channel at a receiver leads to a collision and no packets involved in it can be received correctly by the receiver.

B. Approximate Throughput of HRMA

Let the length of HR, RTS or CTS be γ seconds and the size of the synchronizing period be a multiple of γ, $(c-1)\gamma$. Thus the slot size η equals $(c+2)\gamma$. For simplicity, we ignore the synchronizing slot in our comparative analysis and assume that the synchronizing period of a slot is much longer than the sum of RTS, CTS, and HR period.

There are M frequencies (or frequency hops) available, where $M > N$. This is the case for a typical multi-hop packet radio network operating on the ISM bands and using FH radios as described in Section I, where the number of neighbors of each node is usually smaller than the number of available frequencies.

We observe that any node in a given slot can be either transmitting a data packet, or receiving a data packet, or other than the above two, which we call idle. Note that a node in the idle state can transmit or receive RTS or CTS packet. We assume that the system is in stable operation and the steady state exists. Let P_T, P_R, and P_I be the probabilities that a node is transmitting data, receiving data, and idle in a given slot, respectively. Let $\overline{I}$, $\overline{T}$, and $\overline{R}$ denote the average lengths of an idle period, a data transmitting period and a data receiving period, respectively.

We also observe that, for any node, an idle period must be at least one slot in length and must precede every data transmitting or receiving period, because any successful data transmission or reception must follow a successful RTS-CTS exchange. By this observation, an idle period of a node ends at the end of a slot if and only if there is a successful RTS from or to this node in this slot to initiate a new data transmission. Therefore, the probability that a node ends its idle period during a slot, denoted by q_I, is the probability that the node successfully transmits or receives an RTS in that slot (which are mutually exclusive events) given that it is in the idle state, denoted by P_{STRTS} and P_{SRRTS}, respectively, i.e.,

$$q_I = P_{STRTS} + P_{SRRTS} \tag{1}$$

Denote by $P_{CF|R}$ the probability that a node is on the current frequency hop given that it is receiving a data packet and denote by $P_{CF|T}$ the probability that a node is on the current frequency hop given that it is transmitting a data packet. Let us number the slots from the slot just before the current slot back to all the passed slots as slot 1, 2, ..., and denote the probability that a data transmission is initiated during slot i by $P(i)$. The probability that a data transmission remains in the current slot given that it is initiated in slot i is denoted by $P(T|i)$. Due to the geometric distribution of the packet length, we have $P(T|i) = p^{i-1}$. Because there is no difference between any slot except that different frequency hops may be used for RTS/CTS exchange in different slots, it should hold true that, for any i and j, $P(i) = P(j)$. It follows that

$$P_{CF|T} = \frac{\sum_{j=1}^{\infty} P(T|jM)P(jM)}{\sum_{i=1}^{\infty} P(T|i)P(i)} = \frac{p^{M-1}q}{1-p^M} \tag{2}$$

Let P_{HR} be the probability that a node sends HR packet and P_X be the probability that a node will continue transmitting data on the current frequency hop in the next slot. It yields that

$$P_{HR} = pP_R P_{CF|R} \tag{3}$$

and

$$P_X = pP_T P_{CF|T} \tag{4}$$

The probability that a node has packet arrival is given by $P_A = 1 - e^{-\lambda c\gamma}$. To keep the analysis tractable, we assume that the transmissions of all the neighbors of any given idle node are independent on each other and the given idle node. Denote by P_{RTS} the probability that a node sends RTS for a new data packet. We have

$$P_{RTS} = P_I P_A (1 - P_{HR})^{N-1} \tag{5}$$

Given that a node is idle in a given slot, the node can successfully transmit an RTS to one of its neighbors in the given slot if and only if in the given slot (1) the node has packet arrival during the access period (i.e., the synchronizing or HR period); (2) none of its neighbors other than the destination sends HR, i.e., they will not continue receiving data on the current frequency in the next slot; (3) none of its destination's other neighbors sends RTS, attempting to initiate a new data transmission or to continue transmitting data on the current frequency hop in the next HRMA slot; (4) its destination is idle and does not send RTS. The probability that the destination does not send RTS is the probability that it has no packet arrival, or it has packet arrival but at least one of its neighbors sends HR. We obtain, therefore,

$$P_{STRTS} = P_A(1-P_{HR})^{N-1}(1-P_{RTS}-P_X)^{N-1}P_I \times \left\{(1-P_A) + P_A\left[1 - \left(\frac{1-P_{HR}-P_{RTS}-P_X}{1-P_{RTS}-P_X}\right)^{N-1}\right]\right\}$$

The above equation can be simplified to

$$P_{STRTS} = P_{RTS}\left[(1-P_{RTS}-P_X)^{N-1} - P_A(1-P_{RTS}-P_X-P_{HR})^{N-1}\right] \quad (6)$$

Due to the symmetry of the network topology and the traffic model for the whole system, it is easy to see that $P_{CF|T} = P_{CF|R}$, $P_{STRTS} = P_{SRRTS}$ and $P_T = P_R = (1-P_I)/2$.

Because HRMA guarantees that no data packet collides with other packets, the data transmitting or receiving period has the same distribution as that of the data packet length; therefore $\overline{T} = \overline{R} = \eta/q$.

The duration of the idle period can be modeled as a geometrically distributed random variable with a probability of ending in a slot being q_I. Therefore, we have $\overline{I} = \eta/q_I$.

The idle probability P_I can be calculated as

$$P_I = \frac{\overline{I}}{\overline{T}+\overline{I}} = f(P_I) \quad (7)$$

We now have a set of nonlinear equations in P_I, which can be solved by iteration $P_{I\ new} = P_{I\ old}$. This procedure can converge with only a small number of iterations.

Finally the throughput of HRMA is given by

$$S = P_T = P_R = \frac{1-P_I}{2} \quad (8)$$

C. Approximate Throughput of Multichannel Slotted ALOHA

Prior MAC protocols based on slow frequency hopping assume ALOHA or slotted ALOHA access to the channel and typically assume ROCA (e.g., Metricom's system[2]). There are three type of saturations that can cause collision in the ad-hoc networks with half-duplex single channel radios: (1) multiple packets are directed to the same destination; (2) multiple packets not addressed to the same destination both arrive at any one of the destinations; and (3) the destination is transmitting. In this paper, we assume that a perfect ROCA technique is used for ALOHA, where each node is assigned a frequency such that no two nodes with a same neighbor are assigned a same frequency. A node tunes its transmitter to the frequency assigned to the intended receiver when it needs to transmit its packet. With this assumption, we can eliminate the collisions caused by situation (2) above. In practice, this is not easy to be implemented in a mobile environment. Thus, the throughput we obtain is an upper bound. We assume that transmission preempts any reception. When a packet arrives at a node not transmitting, it will be transmitted at the beginning of the next slot.

For any given node (on a specific frequency) we can construct a queueing system with N customers and N servers. Thus any arrival can get served at the beginning of the next slot. The arrival probability at each idle node in any slot is $p_a = 1-e^{-G}$. The service time for each arrival is the packet length. We can use the Markov model shown in Fig. 5 to describe the operation of this queueing system, where each state of the chain represents the number of transmitting neighbors of the given node (busy servers) during a slot. Let π_k denote the probability that the system is in state $k, 0 \le k \le N$. According to our assumptions, each state of the Markov chain can transit to any state (self loops

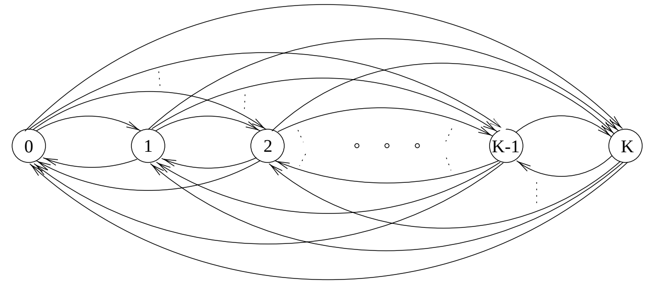

Fig. 5. Markov process for multichannel slotted ALOHA

are omitted in the diagram). A transition may occur in the next slot when any neighbor node finishes transmitting or has packet arrival in a slot.

Denote by $A_k^{(i)}$ and $D_k^{(j)}$ the probabilities that i nodes have packet arrivals and j nodes finish transmitting during a slot in state k, respectively. We have

$$A_k^{(i)} = \binom{N-k}{i} p_a^i (1-p_a)^{N-k-i} \quad 0 \le i \le N-k \quad (9)$$

and

$$D_k^{(j)} = \binom{k}{j} q^j (1-q)^{k-j} \quad 0 \le j \le k \quad (10)$$

For the transition from state k in slot t to state l in slot $t+1$, at least $\hat{n} = max(0, k-l)$ nodes must finish transmitting in slot t. Therefore, the transition probability from any state k to any state l is given by

$$p_{lk} = \sum_{n=\hat{n}}^{k} A_k^{(n+l-k)} D_k^{(n)} \quad (11)$$

We can obtain the state probabilities by solving the global balance equations:

$$\pi_l = \sum_{k=0}^{N} \pi_k p_{lk} \quad 0 \le l \le N$$

with condition $\sum_{l=0}^{N} \pi_l = 1$.

Denote by $B_i^{(j)}$ the probability that j nodes are sending packets to a given node given that i nodes are transmitting. $B_i^{(j)}$ can be expressed as

$$B_i^{(j)} = \binom{i}{j} \left(\frac{1}{N}\right)^j \left(\frac{N-1}{N}\right)^{i-j} \quad 0 \le j \le i \quad (12)$$

The probability for any node R to successfully receive a data packet is equal to the probability that: (a) only one packet is directed to node R from its neighbors in a slot, (b) any packet currently being transmitted to or from node R (if any) ends during this slot, and (c) no any other packet will be transmitted to or from node R during its receiving time. To keep the analysis tractable, we assume that, during the receiving time of any packet at R, any neighbor of R can at most send one packet, which leads to an upper bound of the throughput because we underestimate the collision probability. The probability that an

idle neighbor of node R has no packet arrival for R in i consecutive slots, denoted by E_i, is

$$E_i = 1 - \frac{1 - e^{-iG}}{N} \tag{13}$$

It follows that the probability of (c), when R has r idle neighbors, is

$$C_r = \sum_{s=1}^{\infty} p^{s-1} q (1 - p_a)^{s-1} E_{s-1}^{r}. \tag{14}$$

Therefore, the throughput for any node that is not transmitting when the packet arrives is given by:

$$S_1 = \sum_{k=0}^{N-1} \pi_k \sum_{m=1}^{N-k} A_k^{(m)} B_m^{(1)} \sum_{j=0}^{k} B_k^{(j)} q^j \sum_{n=0}^{k-j} D_{k-j}^{(n)} C_{N-k-m+j+n} \tag{15}$$

In any slot, a node is either transmitting or not transmitting. We can use a simple two-state Markov chain with p_a as the transition probability from nontransmitting state to transmitting state and q as the transition probability for the reverse direction to describe its behavior. Solving this Markov chain, we get the transmitting probability for any node in a slot $P_t = p_a/(p_a + q)$. Therefore, the throughput of multichannel slotted ALOHA is given by

$$S = (1 - P_t) S_1 + P_t q S_1 \tag{16}$$

V. Numerical Results

The numerical results are given in Fig. 6 through Fig. 10, which depict the throughput per node S as a function of normalized offered load per node G with different numbers of neighbors per node N, different values of average packet length APL, or different numbers of frequencies available M (for the case of HRMA), to reflect the effect of different choices of the network parameters on the performance.

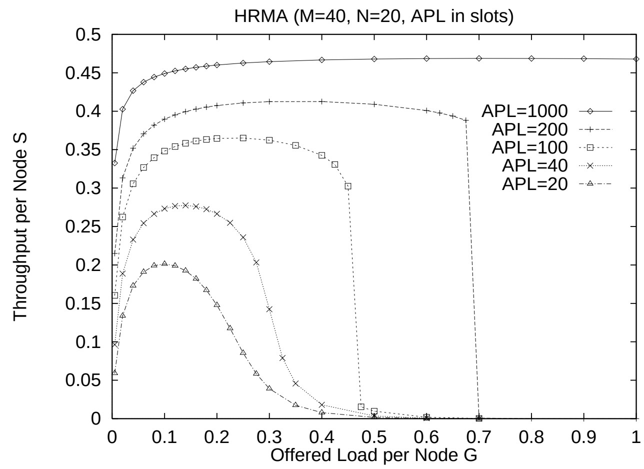

Fig. 6. Throughput of HRMA with different values of average packet length

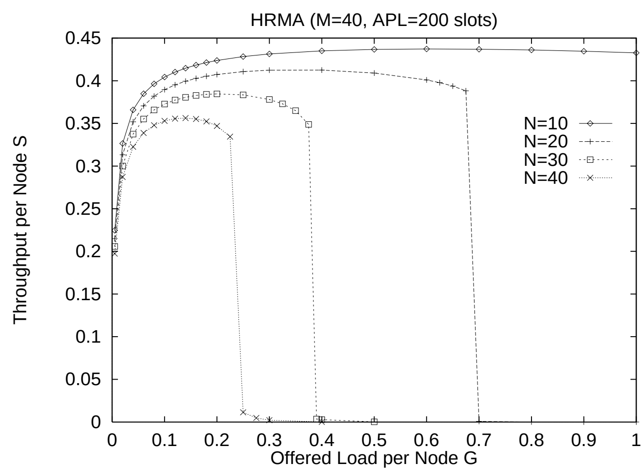

Fig. 7. Throughput of HRMA with different numbers of neighbors per node

Fig. 6 plots the throughput of HRMA with different values of average packet length in slots, where each node has 20 neighbors and there are 40 frequencies available. This is a typical configuration of ad-hoc networks operating on 2.4GHz ISM band using FHSS radios. Throughput grows significantly when the APL increases, with the maximum throughput being close to the theoretical maximum value. This is because HRMA eliminates data collisions; once successful, the large data packets can reserve the frequency for a long time; thus greatly reducing the overhead and improving the utilization of the channel. Also notice that the range of traffic load within which the network stays stable becomes larger with larger APLs. HRMA is more attractive with large packet or packet train.

The throughput of HRMA with average packet length of 200 slots and 40 available frequencies is displayed in Fig. 7 for systems with 10, 20, 30 and 40 neighbors per node. The curves indicate that the throughput and the range of traffic load within which the network stays stable increases as the number of neighbors per node decreases. This is expected, because HRMA uses the common signaling channel (current hop), more neighbors per node leads to more collisions in the signaling channel.

In Fig. 8, we show the effect of changing the number of available frequencies on the throughput. Each node of the system has 10 neighbors and APL=80 or 40 slots. The number of frequencies available has little effect on the performance. HRMA allows idle nodes to contend for the frequency on every unreserved hop by sending RTSs on the common hop. As long as the number of available frequencies is no less than the number of neighbors per node, the success probability for RTS will not change much with additional frequencies. Again, we see that the APL plays a very important role on the performance. Systems in our examples with the same APL almost show the same throughput, even with the different numbers of frequencies available.

Fig. 9 shows the throughput performance for the multichannel slotted ALOHA with different values of APL and different numbers of neighbors per node. Larger APL leads to more collisions and thus lowers the throughput. The throughput for ALOHA is very low even if the APL is small, e.g., when APL=2 slots, the maximum throughput is less than 0.08. Also, APL is the most important factor affecting the throughput, while the number of neighbors per node has little effect on the throughput.

Fig. 10 compares the throughput performance of HRMA and slotted ALOHA with perfect ROCA for the systems where each node has 20 neighbors and 40 frequencies are available for HRMA. APLs are 200 slots and 40 slots for HRMA, and 2 slots and 4 slots for ALOHA. Since the throughput of HRMA in-

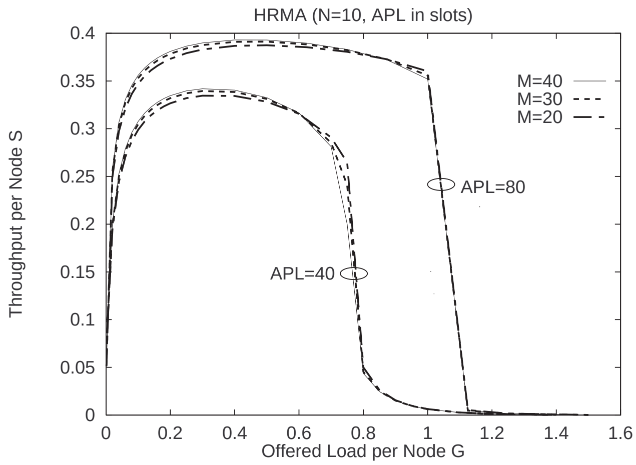

Fig. 8. Throughput of HRMA with different numbers of frequencies

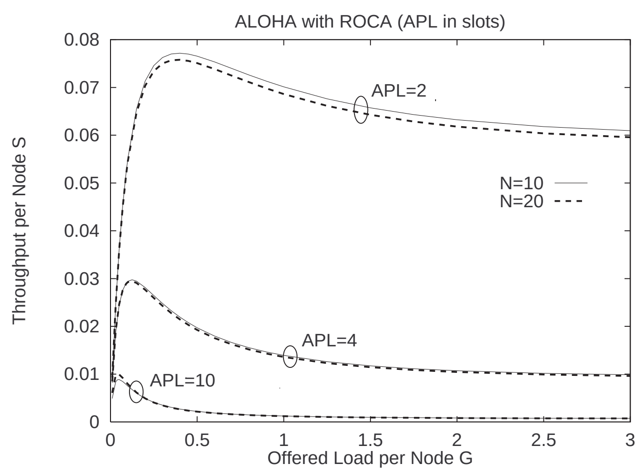

Fig. 9. Throughput of ALOHA with different numbers of neighbors per node and different values of APL

creases when APL increases and HRMA is intended for systems with packet size larger than the frame size, we choose APL=40 slots as the worst-case parameter. While for ALOHA, the throughput decreases when APL increases, thus we choose APL=2 slots as the best-case parameter. It can be seen that in the traffic-load range of interest and with large average packet length compared to the slot size, HRMA performs much better than ALOHA. Moreover, HRMA has the potential to get close to the theoretical maximum performance value with very large packet size, and the throughput of HRMA can be improved further with more sophisticated backoff algorithms or collision resolution algorithms, for example.

VI. CONCLUSIONS

We have described a new multichannel MAC protocol for ad-hoc networks (multi-hop packet radio networks) operating with simple FHSS radios on ISM bands and analyzed its performance. HRMA dynamically allocates frequency bands to nodes using a common frequency-hopping pattern, such that data and acknowledgements are transmitted without hidden-terminal interference. HRMA allows systems to merge and nodes to join existing systems. HRMA's features are achieved using simple half-duplex slow frequency-hopping radios without carrier sensing, which are commercially available today. Our analysis shows that HRMA's throughput performance is significantly better than slotted ALOHA with perfect ROCA, which is representative of the current practice using commercial radios, and that HRMA can achieve a maximum throughput that is comparable to the theoretical maximum value, especially when data packets are large compared to the slot size used for frequency hopping. This high throughput is obtained through a very simple reservation mechanism without the need for complex code assignment.

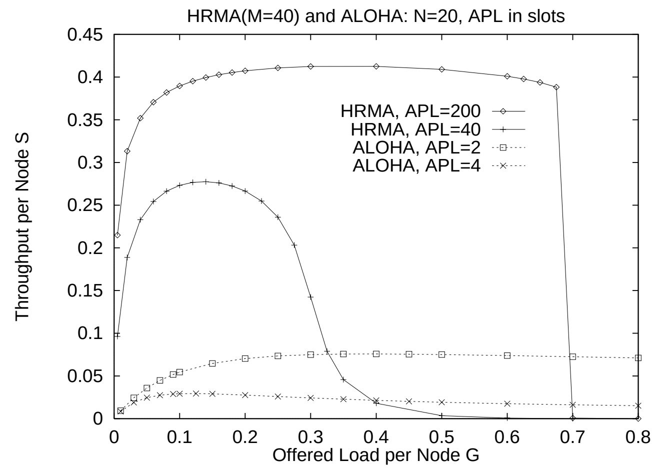

Fig. 10. Throughput comparison: HRMA and ALOHA

REFERENCES

[1] *IEEE P802.11 Draft Standard for Wireless LAN: Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE, July 1996.
[2] *Metricom*. http://www.metricom.com, Los Gatos, CA, 1998.
[3] *FCC Rules and Regulations, Part15.247 and 15.249*. October 1997.
[4] *WINGs for the Internet project and SPARROW project*. http://www.cse.ucsc.edu/research/ccrg/, Santa Cruz, CA, 1998.
[5] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," *Proc. ACM SIGCOMM '97*, Cannes, France, September 14-18, 1997.
[6] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," in *Proc. ACM SIGCOMM '94*, pp. 212–25, London, UK, Aug. 31–Sept. 2, 1994.
[7] A. Ephremides, J. Wieselthier, and D. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling,"*Proceedings of the IEEE*, Jan. 1987, vol.75, (no.1):56-73.
[8] M. Pursley, "The role of spread spectrum in packet radio networks,"*Proceedings of the IEEE*, Jan. 1987, vol.75, (no.1):116-34.
[9] D. Bertsekas and R. Gallager, *Data Networks*. 2nd Edition, Prentice-Hall, 1992.